

# ***EXPERIMENT Cisco MPLS-L3VPN Dengan GNS3***

**Rafdian Rasyid**  
*rafedianr@gmail.com*

## ***Lisensi Dokumen:***

*Copyright © 2003-2008 IlmuKomputer.Com*

*Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarluaskan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.*

## **DAFTAR ISI**

<b>Membuat LAB <i>Networking</i> US\$210.000 .....</b>	<b>2</b>
<b>GNS3 – Menyempurnakan <i>Dynamips</i> .....</b>	<b>2</b>
<b>BAGIAN I : SEKELUMIT TENTANG GNS3 .....</b>	<b>5</b>
<b>Dynamips dan Dynagen.....</b>	<b>5</b>
<b>BAGIAN II: EXPERIMENT MPLS-L3VPN.....</b>	<b>7</b>
<b>PERTAMA. CONFIGURE SEMUA INTERFACE .....</b>	<b>13</b>
<b>KEDUA. AKTIFKAN DYNAMIC ROUTING.....</b>	<b>13</b>
<b>KETIGA. AKTIFKAN BGP.....</b>	<b>14</b>
<b>KEEMPAT. AKTIFKAN MPLS.....</b>	<b>16</b>
<b>KELIMA. BUAT ROUTER VIRTUAL .....</b>	<b>17</b>
<b>KEENAM. MENAMBAHKAN ROUTING DI ROUTER VIRTUAL.....</b>	<b>18</b>
<b>KETUJUH. AKTIFKAN MP-BGP.....</b>	<b>19</b>
<b>KEDELAPAN. MENGAKTIFKAN CE dan MENGUJI LINK END-To-END ...</b>	<b>22</b>
<b>Biografi Penulis .....</b>	<b>23</b>
<b>KONFIGURASI.....</b>	<b>24</b>
PE1 .....	24
PE2 .....	26
Core.....	28
VPN1_A.....	30
VPN2_X.....	31
VPN1_B.....	33
VPN2_Y.....	34

## **Membuat LAB Networking US\$210.000**

Tiba-tiba Anda diberikan 7 buah Cisco Router 7200, seharga total US\$210 K (atau sekitar dua milyar rupiah). Dan itu Anda peroleh secara cuma-cuma! “Wow...!!!”.

Kejadian diatas, sekarang ini, sangatlah mudah untuk Anda alami dan dapatkan. *Dynamips* + *Dynagen* + *GNS3*, telah memungkinkan Anda, menjadi milyarder dalam sekejap! Anda bahkan bisa membuat 10 atau 20 Cisco Router 7200 (tergantung kekuatan PC Anda), dan Anda bisa membuat ATM Switch, Frame-relay Switch, dan banyak lagi. Bisa-bisa Anda dapat membuat sebuah Lab pribadi seharga 10 milyar rupiah lebih... Wow!!!

## **Berkenalan dengan Dynamips**

Betapa menyenangkan jika kita dapat “mensimulasikan” beberapa Router Cisco lalu kita jajah pengetahuan kita tentang Routing dan Switching. Sebutlah mencoba-coba BGP, OSPF dsb.

Saya pakai kata “mensimulasikan” karena yang akan kita lakukan adalah mensimulasikan Cisco menggunakan aplikasi Dynamips. Aplikasi ini open-source dan freeware, alias gratis.

Memiliki dan dapat ngoprek Cisco Router / Switch adalah satu kesenangan sendiri bagi orang yang gemar terhadap networking. Saya masih ingat dulu waktu hendak ujian CCNP, sekitar 8 tahun yang lalu. Ada 4 ujian yang harus dilewati, salah satunya Materi tentang Routing & Switching. BGP dan OSPF adalah fokus yang diujikan. Rasanya kalau tidak bekerja ditempat yang punya Router Cisco 3600 dan 7200 akan sukar untuk bisa dengan cepat memahami materi BGP. Ya, percobaan dengan “real” router, sangat mempercepat pemahaman suatu konsep.

Banyak simulator Router Cisco yang sudah dibuat orang, tapi satupun tidak ada yang memuaskan. Mengapa? Router simulator yang dibuat tersebut biasanya sangat tidak presisi, command terbatas, dan tidak bisa bekerja seperti layaknya sebuah IOS. Kalau di *rating*, yaah, paling simulator Router Cisco hanya bisa menyamai 10% dari kemampuan “real” router.

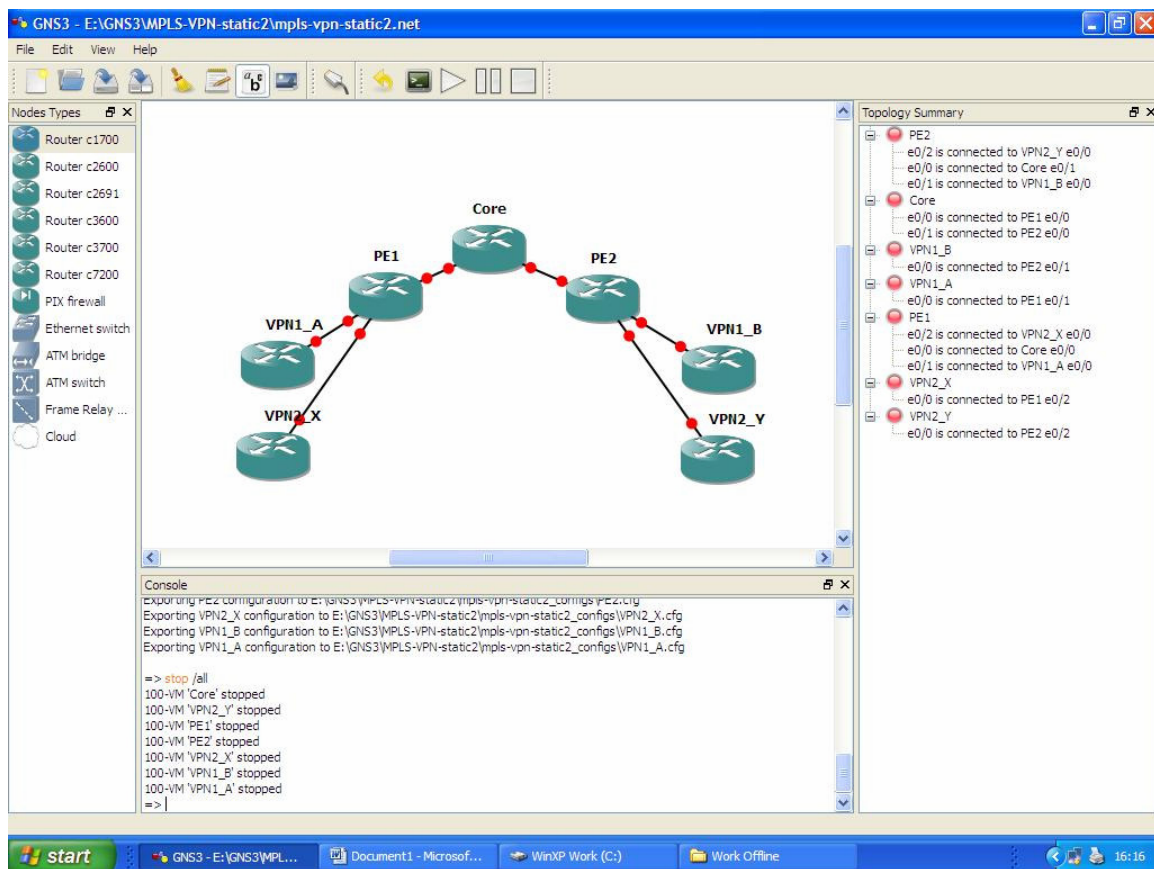
Sudah lama saya membayangkan, mengapa tidak ada emulator IOS, atau lebih pasnya sebuah *Virtual Machine* (VM) yang jalan di PC biasa, lalu dengan men-upload real Cisco IOS, diatas VM tersebut maka “real-emulated” Router Cisco kita dapatkan.

## **GNS3 – Menyempurnakan Dynamips**

Saya mendapatkan informasi tentang Dynamips dari seseorang, baru-baru ini. Dia mengatakan bahwa software Dynamips tersebut dapat menjalankan IOS Cisco 7200. “Waw”.... Kata saya dalam hati. Setelah sekian lama tidak ngoprek Cisco lagi, ternyata sekarang sudah ada VM terhadap IOS Cisco? “Great...Fantastic!!!”.

Mulai saya coba aplikasi Dynamips tsb. Berhasil! Tapi saya sedikit kecewa, karena interface GUI tidak tersedia. Dari search di Internet, saya menemukan bahwa GNS3 adalah interface GUI untuk Dynamips dan Dynagen. (catatan: saya tidak akan panjang lebar menjelaskan Dynamips dan Dynagen ini karena bukan di artikel ini tempat pembahasannya)

Oke, kembali ke GNS3, untuk merasakan “taste” dari GNS3, berikut ini tampilan GUI nya:



Gambar 1. Tampilan GNS3

Terlihat bahwa secara tampilan, aplikasi GNS3 ini mirip dengan Simulator Cisco pada umumnya. Apa hebatnya? Yah seperti saya katakan, Cisco Router yang ada dalam gambar diatas **benar-benar** menggunakan IOS dari router yang sebenarnya. Sehingga bisa dipastikan semua akan mensimulasikan real-world. “Wow keren boo...”.

Saya akan bagi tulisan ini menjadi 2 bagian. Bagian I mengenai "Sekelumit tentang GNS3", dan Bagian II adalah "Experiment MPLS-L3VPN".

## **BAGIAN I : SEKELUMIT TENTANG GNS3**

Saya tidak akan jelaskan bagaimana cara instalasinya GNS3. Karena seperti Anda ketahui instalasi sebuah Paket aplikasi diatas MS Windows biasanya gampang sekali. Anda tinggal menklik Ok, Ok, dst.

Aplikasi GNS3 ini dapat Anda download di : <http://www.gns3.net/download>

Ambillah versi yang sudah all-in-one, yaitu yang sudah berisi GNS3, Dynamips, dan Dynagen. Waktu tulisan ini dibuat versi yang digunakan adalah **GNS3-0.5-win32-all-in-one.exe**

### **Dynamips dan Dynagen**

Anda tidak usah terlalu memusingkan apa dan bagaimana cara penggunaan Dynamips dan Dynagen (walau Anda masih perlu tahu command-command Dynagen yang penting, seperti *start*, *stop*, *list*, *suspend*, *resume*, *telnet*). GNS3 telah mengintegrasikan kedua aplikasi itu. Anda tinggal melakukan setting terhadap Dynamips dari menu-menu yang sudah disediakan GNS3.

Beberapa parameter yang **paling** penting Anda set di GNS3:

Edit → Preferences → Dynamips :

- contreng: Enable ghost IOS feature
- contreng: Enable mmap feature
- contreng: Enable sparse memory feature

Tiga contrengan (*check mark*) diatas itu sangat-sangat-sangat penting untuk Anda kerjakan. Percayalah jika tidak Anda contreng, PC Anda akan bekerja dengan lambaaaaaat sekali.

Ada lagi satu bagian, yaitu *idle-pc* yang juga Anda harus set, pada IOS yang Anda import ke GNS3. Parameter *idle-pc* ini unik untuk tiap IOS. Sekali Anda ketemu nilai yang pas, Anda save. Tiap Anda menambahkan satu Router baru dalam diagram GNS3, maka otomatis nilai *idle-pc* itu akan digunakan untuk router dengan IOS tsb. Kalau *idle-pc* tidak Anda set, maka PC Anda akan bekerja dengan lambaaaaaat sekali.

Mengapa? Karena Dynamips mengemulasikan sebuah Router, dimana *idle\_process* pada Cisco Router berbeda dengan *idle\_process* pada CPU PC. Sehingga jika Dynamips jalan (saat Anda men – start – sebuah Router di GNS3) maka Dynamips akan memakan 100% dari CPU PC process, walau router dalam keadaan idle. Untuk bisa men-drop CPU PC utilization dari 100% turun sampai 1 – 10%, maka Dymamips perlu diberitahu mengenai nilai *idle-pc* dari IOS Cisco yang digunakan.

Oke, jika Anda ingin tahu cara setting idle-PC silahkan baca manual dari GNS3 atau manual dari Dynamips.

Bagian I ini tidak boleh panjang-panjang, karena fokus saya bukan dibagian I ini. Silahkan mencari referensi lain, untuk instalasi dan setting GNS3.

Kita segera masuk kebagian II, yaitu bagian utama yang hendak kita bahas, di halaman berikut ini.

## **BAGIAN II: EXPERIMENT MPLS-L3VPN**

Di bagian inilah yang saya akan coba jelaskan setahap demi tentang bagaiman caranya ber-experiment dengan MPLS-L3 VPN. Apa asumsi yang saya buat terhadap profile pembaca? Saya berasumsi Anda sudah mengerti bagaimana:

- cara setup Router Cisco secara sederhana, dengan routing statik
- cara setup salah satu dynamic routing di Router Cisco, minimal RIP

Jika Anda sudah mengerti dua hal diatas, maka experimen berikut ini akan meningkatkan pengetahuan Anda tentang:

- Fungsi BGP, cara setup BGP, dan verifikasinya
- Fungsi OSPF, cara setup OSPF, dan verifikasinya
- Konsep jaringan MPLS, dan cara setup serta verifikasinya
- Cara setup router virtual dengan VRF
- Cara setup MP-BGP
- Cara menyuntikkan routing prefix ke MP-BGP

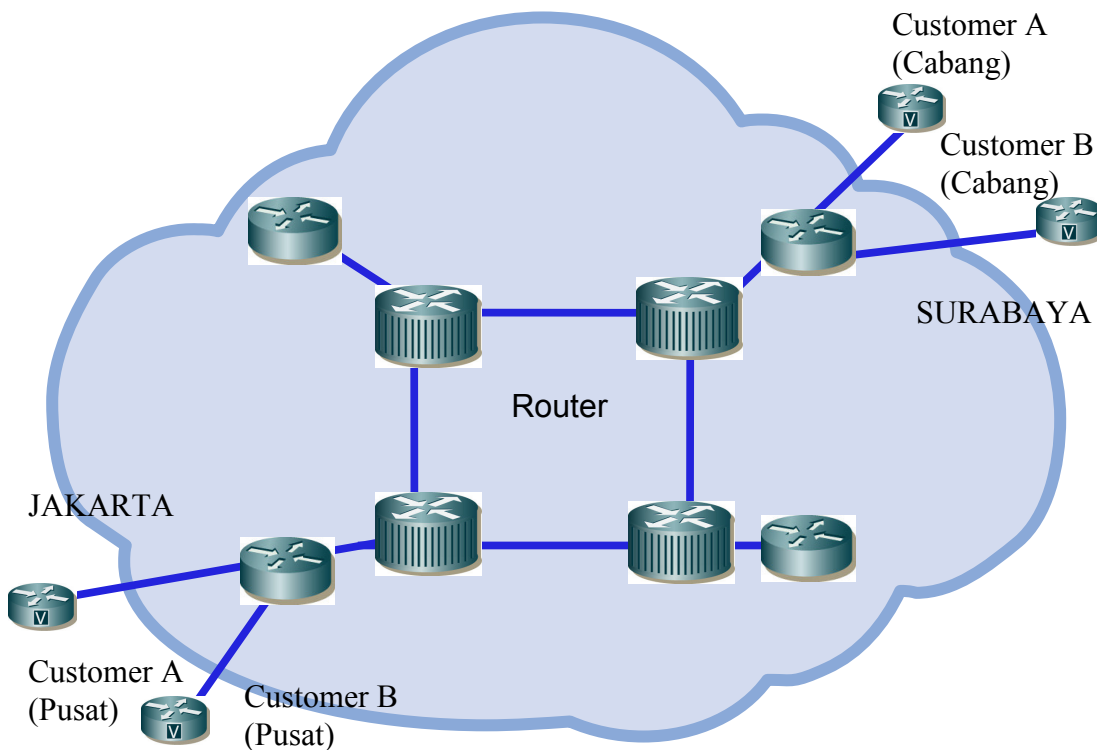
Kita mulai dengan membayangkan skenario, sbb:

Anda akan mendirikan sebuah perusahaan yang menyediakan layanan Jaringan MPLS. Tahap pertama sudah ada 2 customer yang mendaftar.

Customer A: Lokasi Jakarta dan Surabaya

Customer B: Lokasi Jakarta dan Surabaya

Inilah bentuk Network MPLS Anda:



Gambar 2. Konfigurasi awal

Kebutuhan Customer A dan B, sederhana sekali, yaitu mereka ingin koneksi (IP Connection) dari kantor Pusat Jakarta ke kantor Cabang mereka di Surabaya.

Anda mungkin akan mengatakan: gak perlu MPLS, cukup ditengah (yang ada dalam cloud) hanya jaringan IP saja.

*Well*, jawabannya bisa saja begitu. Tapi bagaimana kalau Customer A dan Customer B, sama-sama menggunakan IP prefix yang sama?

Misalkan IP Address yang digunakan di Jakarta

- Customer A: 202.147.192.0/24
- Customer B: 202.147.192.0/24

Nah akan repot bukan?

Eit... kan bisa pakai NAT (Network Address Translation), atau pakai solusi VPN L3 Point-to-point menggunakan enkripsi GRE, atau IPSec, sehingga dalam awan tidak akan terjadi IP yang "bentrok" antara Customer A dan Customer B?

Anda benar. Tapi solusi NAT dan VPN tadi bukanlah solusi yang tepat, karena tidak *scalable*, dan performance router yang menurun dibandingkan dengan IP Route biasa, karena terbebani dengan processing VPN encryption, dsb.

Jadi apa solusinya?

Solusi yang paling tepat adalah MPLS.

Jadi yang Anda lakukan adalah:

→ Mengganti cloud IP di gambar 2 diatas menjadi cloud MPLS.

Bagaimana caranya? Caranya adalah semua router yang ada dalam cloud di gambar 2, Anda upgrade menjadi router yang memiliki kemampuan MPLS.

Bagaimana cara setting atau cara mengconfigure MPLSnya? Nah, ini yang akan saya jelaskan.

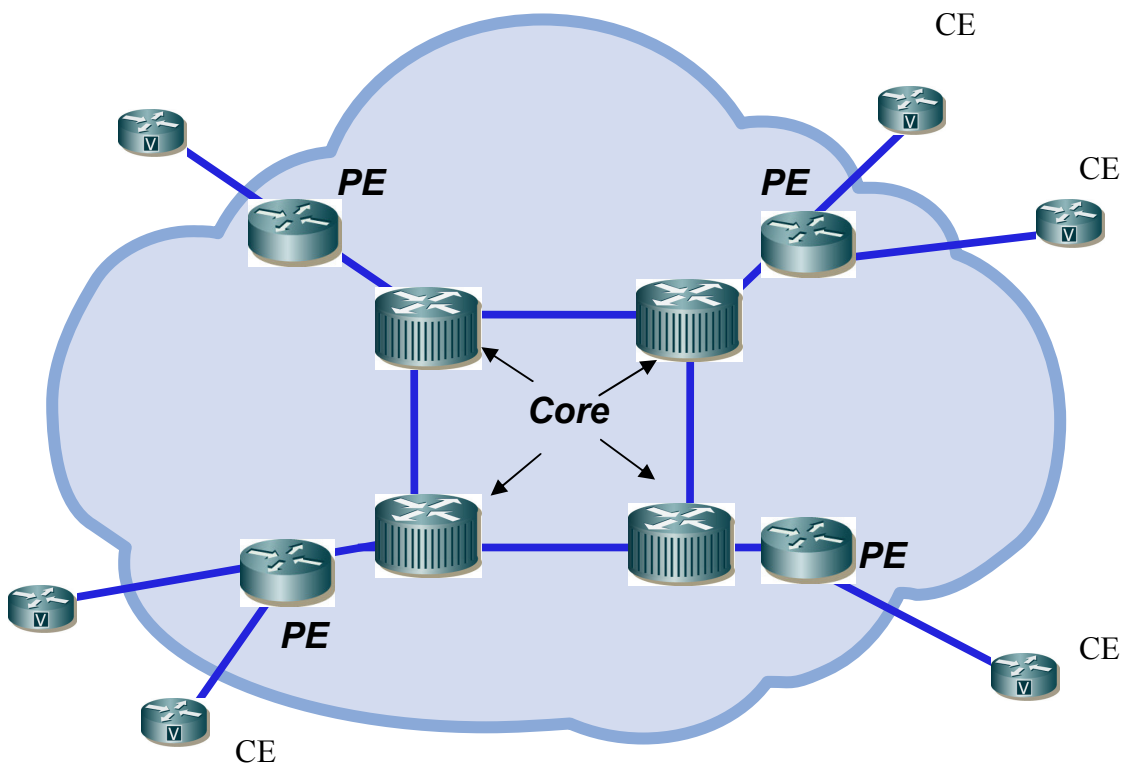
Kita akan pakai GNS3 untuk mensimulasikan masalah ini.

## **KONFIGURASI KITA**

MPLS pada dasarnya terdiri dari 3 jenis router:

1. Router Core
2. Router PE
3. Router CE

Perhatikan gambar 3



Gambar 3. Posisi Router Core, PE, dan CE dalam konsep jaringan MPLS

Jika diserhanakan maka sebuah cloud MPLS terdiri dari 2 lapisan :

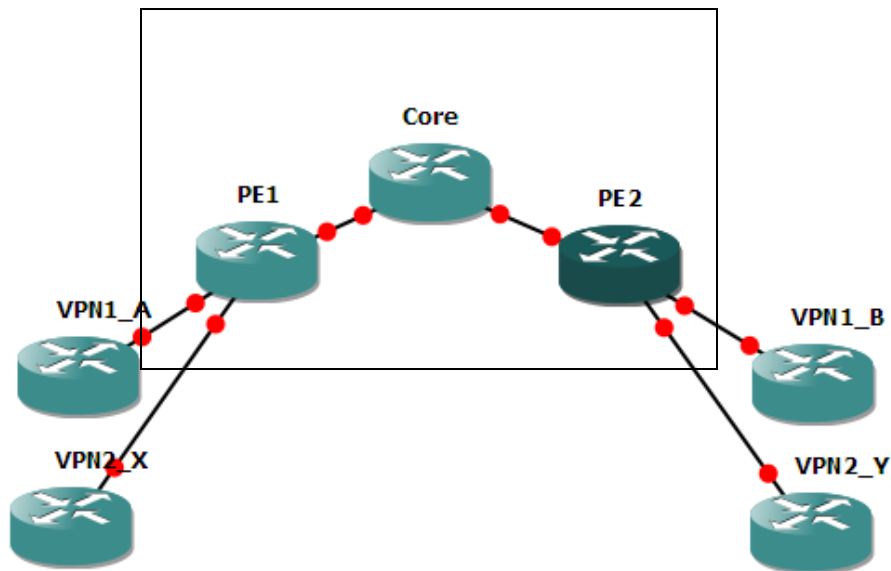
1. Lapisan inti (terdiri dari interkoneksi sesama router Core)
2. Lapisan luar (terdiri dari Router PE yang mengitari lapisan inti)

Antara PE didalam cloud MPLS, harus saling berkomunikasi menggunakan BGP + MP-BGP (nanti akan dijelaskan apa fungsi kedua protokol ini). Pada saat ini, cukuplah diketahui, bahwa jika ingin men-deliver L3 VPN dalam cloud MPLS, maka semua router PE harus saling berkomunikasi menggunakan MP-BGP.

Oleh karena itu satu jaringan MPLS harus ada router Core, dan Router PE, maka model jaringan MPLS yang paling sederhana dapat berbentuk sbb :

1. Lapisan inti hanya terdiri dari 1 Core
2. Lapisan luar yang terdiri dari 2 PE

Hal ini dapat dilihat pada gambar 4, berikut ini.



Gambar 4. Bagian yang ada dalam kotak adalah contoh dari Cloud MPLS yang paling sederhana, terdiri dari 1 Core dan 2 PE

Perhatikan gambar 4 diatas. Ya, gambar ini adalah gambar permasalahan kita. Di dalam gambar tsb ada dua customer :

Customer A, yaitu VPN1 yang ingin menghubungkan Jakarta (Router VPN1\_A) dan Surabaya (Router VPN1\_B)

Customer B, yaitu VPN2 yang ingin menghubungkan Jakarta (Router VPN2\_X) dan Surabaya (Router VPN2\_Y)

Dari mana kita mulai ?

Oke kita mulai dari arah Customer dulu saja.

Tabel 1. Konfigurasi IP pada Router Customer:

Router Name	Loopback 0	Ethernet 0/0
VPN1_A	202.147.192.1/32	192.168.1.1/24
VPN2_X	202.147.192.1/32	192.168.6.1/24
VPN1_B	202.147.192.2/32	192.168.4.2/24
VPN2_Y	202.147.192.2/32	192.168.5.2/24

Perhatikan bahwa di Jakarta, Loopback 0 untuk Router VPN1\_A dan VPN2\_X, adalah **sama**, yaitu 202.147.192.1/32. Jika dalam jaringan IP biasa (bukan jaringan MPLS), router PE1 **pasti akan "bingung"** menghadapi kedua router tsb, yang memiliki IP yang sama.

Bayangkan kalau Anda melakukan PING dari PE2 ke 202.147.192.1. Paket akan sampai ke PE1 lalu kemanakah PE1 harus melempar paket PING tsb, apakah ke router VPN1\_A, atautkah ke VPN2\_X? Dua-duanya meng-klaim memiliki address 202.147.192.1!!!! Pastilah PE1 akan bingung. Nanti akan dijelaskan bagaimana PE1 mengatasi hal ini, yaitu dengan solusi menggunakan VRF (Virtual Routing Forwarding).

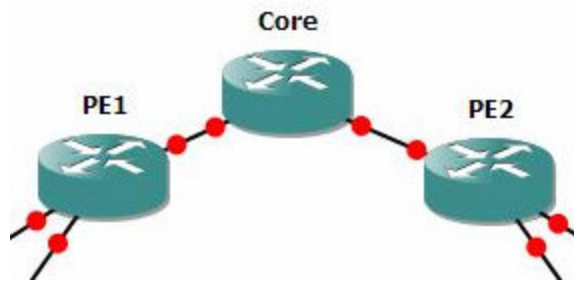
Kembali ke contoh diatas. Kita sudah mengetahui konfigurasi IP dari sisi Customer.

Sekarang, inilah bagian yang paling seru. Kita akan membuat sebuah awan MPLS, step by step.

Siapkan sebuah router Core, dan dua buah Router PE, dalam awan tsb seperti gambar 4, atau lebih jelasnya di gambar 5, dibawah.

Anda coba buat seperti gambar 4, pada program GNS3 di computer Anda. Perhatikan bahwa IOS yang Anda gunakan haruslah IOS yang support MPLS.

Oke. Jika sekarang Anda sudah punya gambar berikut di GNS3 Anda:



Gambar 5. Representasi dari Jaringan MPLS paling sederhana

Maka inilah daftar interface beserta IP Address yang akan digunakan.

Tabel 2. Konfigurasi IP masing-masing Interface pada router Service Provider:

Router Name	Loopback 0	Eth 0/0	Eth 0/1	Eth 0/2
PE1	192.168.100.1/32	192.168.2.1/24	192.168.1.2/24	192.168.6.2/24
Core	192.168.100.2/32	192.168.2.2/24	192.168.3.1/24	
PE2	192.168.100.3/32	192.168.3.2/24	192.168.4.1/24	192.168.5.1/24

Oke. *Ready?*

Mari kita mulai meng-configure semua Router secara urutan.

## **PERTAMA. CONFIGURE SEMUA INTERFACE**

Anda configure semua interface dalam router PE1, PE2, dan Core. Routing static ataupun dynamic belum perlu Anda terapkan pada langkah ini.

Setelah Anda configure semua IP pada interface, cobalah PING router yang interfacenya ada dalam satu network.

Contoh dari PE2, PING ke 192.168.3.1 (Core), jika berhasil maka artinya link antara PE2 dan Core sudah OK.

Demikian juga dari Core Anda PING ke 192.168.2.1 (PE1), jika berhasil maka artinya link antara PE1 dan Core sudah OK.

## **KEDUA. AKTIFKAN DYNAMIC ROUTING**

Sekarang tentulah Anda ingin agar routing dimasing-masing router “mengembang” atau “terpopulasi”. Supaya apa? Supaya dari PE1 bisa PING ke PE2 misalkan. Artinya, ada *connectivity* antara PE1 ke PE2, dan juga antara PE1/PE2 ke Core.

Sampai sejauh ini, Anda tidak akan bisa melakukan PING tsb, sebelum informasi dalam routing-table masing-masing router terisi, atau saling bertukar routing-table.

Bagaimana cara “mengembangkan” atau “mempopulasikan”? Ada 2 cara. Statik atau Dynamic. Saya akan pilih dynamic, karena lebih mudah dan scalable. Dan memang demikianlah jika hendak membuat cloud MPLS, semua router dalam cloud tsb harus menggunakan dynamic routing.

Saya akan pilih OSPF.

Ini adalah konfigurasi OSPF untuk router PE2

```
router ospf 100
 network 192.168.3.0 0.0.0.255 area 100
 network 192.168.100.3 0.0.0.0 area 100
```

Ingat network yang akan diadvertise adalah interface loopback 0 dan network dari ethernet yang directly connected. Dalam konteks PE2, interface yang kita advertise hanya network dari Eth 0/0 yang mengarah ke Core. Sedangkan interface Eth 0/1 yang mengarah ke CE 2 tidak kita advertise. Hal ini agar di dalam cloud MPLS, IP prefix yang di advertise hanyalah yang benar-benar ”didalam” cloud. IP kearah customer tidak di-consider untuk ikut di advertise.

Ini adalah konfigurasi OSPF untuk router PE1

```
router ospf 100
 network 192.168.2.0 0.0.0.255 area 100
```

```
network 192.168.100.1 0.0.0.0 area 100
```

Dan ini adalah konfigurasi OSPF untuk router Core

```
router ospf 100
network 192.168.2.0 0.0.0.255 area 100
network 192.168.3.0 0.0.0.255 area 100
network 192.168.100.2 0.0.0.0 area 100
```

Sesudah selesai dengan mengaktifkan OSPF, sekarang coba Anda perhatikan apakah routing-table Anda sudah “terpopulasikan”.

Caranya:

Di PE1: Anda periksa routing table Anda.

```
PE1# sh ip route
```

Jika ada routing OSPF dari arah PE2, dan Core, maka dapat dipastikan OSPF telah berjalan dengan baik. (catatan: routing OSPF biasanya ditandai dengan “O” pada entry di routing table)

Sekarang Anda bisa melakukan PING, dari PE2 ke loopback 0 PE1

```
PE2# ping 192.168.100.1
```

Jika ini berhasil maka langkah pada tahap ini selesai.

### **KETIGA. AKTIFKAN BGP**

Perlu Anda ketahui, MPLS-VPN menggunakan MP-BGP (Multiprotocol BGP), yaitu sebuah extension dari protocol BGP Standard. Untuk dapat menggunakan MP-BGP, Anda harus punya BGP connection antar PE. Anda akan mempelajari bagaimana meng-configure MP-BGP nanti pada langkah KETUJUHAN. Pada langkah ini kita cukup mengaktifkan BGP dulu.

Dalam case kita, maka BGP connection pertama-tama harus established dulu antara PE1 dan PE2.

Anda configure BGP pada PE1

```
router bgp 100
no synchronization
neighbor 192.168.100.3 remote-as 100
neighbor 192.168.100.3 update-source Loopback0
no auto-summary
```

dan Anda, juga configure BGP pada PE2

```
router bgp 100
  no synchronization
  neighbor 192.168.100.1 remote-as 100
  neighbor 192.168.100.1 update-source Loopback0
  no auto-summary
```

Setelah Anda configure, maka Anda periksa apakah TCP connection antara BGP yang ada di PE1 dan PE2 sudah established, atau belum. Jika tertulis statusnya Active (bukan Established) maka dapat dipastikan bahwa peering BGP belum terbentuk. Anda periksa lagi konfigurasi yang digunakan.

Cara lain untuk memeriksa status BGP, bisa memeriksa disalah satu router.

```
PE1# sh ip bgp neigh
```

Ada banyak informasi yang akan keluar. Yang Anda perlu tahu hanyalah apakah peering (atau bisa disebut tunnel) BGP sudah terbentuk antara PE1 dan PE2.

Anda bisa juga melihat hal tsb dengan perintah berikut.

```
PE1# sh ip bgp sum
```

Perhatikan statusnya, apakah Active, jika active maka connection (BGP peering) belum terbentuk.

Oke. Jika sampai tahap ini BGP connection sudah established, maka langkah selanjutnya adalah, membuat router virtual di PE1 dan PE2.

Mengapa kita perlu router virtual? Karena kita berhadapan dengan customer yang bisa saja IP prefix yang digunakan sama. Identy adalah, bahwa router virtual di PE1 dan PE2 itu adalah "mini-router" yang dedicated untuk mengatur routing table Customer A (vpn1) dan Customer B (vpn2). Karena konsepnya dedicated, maka jumlah router virtual ini akan sebanyak jumlah VPN Customer yang terhubung ke PE itu (dalam case kita ada 2 router virtual didalam pysical router PE1, karena ada 2 customer yang terhubung ke dia).

Masing-masing router virtual (dalam satu physical router PE), memiliki routing table sendiri-sendiri yang saling terpisah. Nantinya Anda akan bisa memeriksa routing table pada router virtual untuk Customer A (vpn1) misalkan:

```
PE1# sh ip ro vrf vpn1
```

Perintah diatas adalah untuk melihat isi routing table sebuah router virtual bernama vpn1 di dalam router PE1. Kita akan mendalami hal ini nanti. Sekarang kita cukupkan langkah ke tiga ini dimana Anda telah berhasil meng-configure BGP antara PE1 dan PE2. Langkah selanjutnya adalah mengaktifkan MPLS di PE1, PE2, dan Core.

#### **KEEMPAT. AKTIFKAN MPLS**

Agar MPLS aktif di Core dan PE1 dan PE2, maka interface yang harus menerapkan (meng-impose) label, mestilah kita aktifkan MPLS nya.

Perintah berikut digunakan untuk mengaktifkan MPLS di PE1.

```
ip cef
mpls label protocol ldp
mpls ldp router-id Loopback0 force
interface Ethernet0/0
  description *** Connection to Core ***
  ip address 192.168.2.1 255.255.255.0
  mpls ip
```

Perintah berikut digunakan untuk mengaktifkan MPLS di PE2.

```
ip cef
mpls label protocol ldp
mpls ldp router-id Loopback0 force
interface Ethernet0/0
  description *** Connection to Core ***
  ip address 192.168.3.2 255.255.255.0
  mpls ip
!
```

Dan terakhir aktifkan MPLS di Core.

```
interface Ethernet0/0
  description *** Connection to PE1 ***
  ip address 192.168.2.2 255.255.255.0
  mpls ip
!
interface Ethernet0/1
  description *** Connection to PE2 ***
  ip address 192.168.3.1 255.255.255.0
  mpls ip
!
```

Setelah MPLS aktif, Anda dapat memastikan bahwa MPLS sudah bekerja, dengan memeriksa:

```
PE1# show mpls ldp neigh
```

Anda akan melihat apakah PE1 mengenal Core sebagai router MPLS, atau tidak. Jika sudah terbentuk, maka protocol LDP akan saling bertukar MPLS lable table.

## KELIMA. BUAT ROUTER VIRTUAL

Pada langkah KETIGA, Anda telah diperkenalkan dengan konsep router virtual. Bagaimana cara meng-configure-nya? Langkah berikut akan menjelaskan prosesnya.

Untuk membuat router virtual di PE1 dan PE2, Anda menggunakan perintah sbb ini. Contoh untuk PE1 (hal yang sama anda terapkan pada PE2).

```
PE1# sh run
ip vrf vpn1
  rd 100:1
  route-target export 100:1
  route-target import 100:1
!
ip vrf vpn2
  rd 100:2
  route-target export 100:2
  route-target import 100:2
```

Perintah `export` di PE1, artinya semua routing table (atau disebut routing prefix) di router virtual vrf vpn1, jika diadvertise ke PE2, maka akan diberi tanda dengan RD (route distinguisher) 100:1.

Routing prefix tersebut setelah sampai di PE2, akan digunakan. Tapi untuk dapat menggunakannya PE2 harus meng-`import` dulu route dengan RD 100:1 tsb.

Itulah routing untuk Customer A (rd 100:1 vpn1). Sedangkan routing untuk customer B, menggunakan rd 100:2 (untuk vpn2)

Perhatikan bahwa nama vpn1 dan vpn2 berlaku / dikenal secara local di router itu saja. Yang dipertukarkan adalah rd 100:1 dan 100:2. Nomer rd ini akan dipertukarkan antar PE, jadi harus lebih berhati-hati dengan penomoran RD ini.

Setelah Anda masukkan konfigurasi diatas, maka dalam PE1 dan PE2 akan tercipta 2 router virtual yaitu yang ber-id **vrf vpn1** dan **vrf vpn2**.

Sekarang, router virtual itu belum punya interface. Maka kita akan apply router virtual tsb ke interface yang mengarah ke Customer.

Contoh di PE1

```
interface Ethernet0/1
  description *** Connection to VPN1_A***
  ip vrf forwarding vpn1
  ip address 192.168.1.2 255.255.255.0
```

```
!  
interface Ethernet0/2  
  description *** Connection to VPN2_X ***  
  ip vrf forwarding vpn2  
  ip address 192.168.6.2 255.255.255.0
```

Contoh di PE2

```
interface Ethernet0/1  
  description *** Connection to VPN1_B ***  
  ip vrf forwarding vpn1  
  ip address 192.168.4.1 255.255.255.0  
!  
interface Ethernet0/2  
  description *** Connection to VPN2_Y ***  
  ip vrf forwarding vpn2  
  ip address 192.168.5.1 255.255.255.0
```

Nah, sekarang, kita sudah punya interface connected dalam masing-masing router virtual tsb. Kita bisa periksa apakah benar sudah ada interface di routing table router virtual kita.

```
PE1# sh ip ro vrf vpn1
```

Maka sekarang akan muncul satu interface Connected, di routing table vrf vpn1

Selanjutnya, idenya adalah bahwa kita harus bisa "mempopulasi / mengembangkan" routing table masing-masing router virtual.

Kenapa?

Kasusnya mirip dengan awal cerita kita pada langkah PERTAMA, dimana satu router tidak mengenal routing prefix router didepan / dibelakangnya. Artinya satu router tidak bisa PING ke router yang bukan directly connected ke dia. Supaya bisa, maka routing-table harus saling dipertukarkan.

Oke. Sekarang kita sudah punya router virtual, dan sudah punya satu interface Connected dalam routing table router virtual. Sebelum mempertukarkan isi routing-table dari router virtual, maka kita tambahkan dulu routing static yang perlu untuk masing-masing router virtual.

## **KEENAM. MENAMBAHKAN ROUTING DI ROUTER VIRTUAL**

Kalau kita lihat, router virtual kita baru punya satu baris routing table. Dan kita ingat bahwa router virtual ini akan berhadapan dengan CE. Oleh karena kita sudah memutuskan antara PE (lebih tepatnya antara router virtual dalam PE) ke CE menggunakan routing static, maka kita perlu menambahkan routing static kearah CE yang directly connected tsb.

Perintah berikut menambahkan routing static ke arah VPN1\_B dan VPN2\_Y, di PE2

```
ip route vrf vpn1 202.147.192.2 255.255.255.255 192.168.4.2
ip route vrf vpn2 202.147.192.2 255.255.255.255 192.168.5.2
```

Perintah berikut menambahkan routing static ke arah VPN1\_A dan VPN2\_X, di PE1

```
ip route vrf vpn1 202.147.192.1 255.255.255.255 192.168.1.1
ip route vrf vpn2 202.147.192.1 255.255.255.255 192.168.6.1
```

Setelah Anda tambahkan, Anda dapat memeriksa tambahan routing tsb di dalam router virtual.

```
PE1# sh ip ro vrf vpn1
PE1# sh ip ro vrf vpn2
```

Anda akan melihat bahwa sekarang di router virtual (misalkan vrf vpn1) akan ada 3 baris routing, satu Connected, dan 2 baris untuk Static.

Nah, ini langkah terakhir. Bagaimana caranya routing table di router virtual (misalkan vrf vpn1) yang ada di PE1 dapat di "export" ke PE2, agar network di PE1 router virtual vrf vpn1 bisa masuk di routing table di PE2 router virtual vrf vpn1?

Langkah ini akan dijelaskan sbb.

## **KETUJUH. AKTIFKAN MP-BGP**

Anda masih ingat di langkah KETIGA, yaitu kita sudah mengaktifkan BGP antara PE1 dan PE2. Anda bisa membayangkan "tunnel" BGP ini akan berisi "sub-tunnel" MP-BGP untuk membawa routing table vrf vpn1 dari PE1 ke PE2. Demikian juga dengan sebaliknya. Dan "sub-tunnel" MP-BGP ini juga akan membawa routing table vrf vpn2 dari PE1 ke PE2. Demikian juga dengan sebaliknya.

Begini caranya Anda mengaktifkan MP-BGP di PE1.

```
router bgp 100
  address-family vpnv4
  neighbor 192.168.100.3 activate
  neighbor 192.168.100.3 send-community both
  exit-address-family
!
```

Langkah diatas mengatakan, seolah-olah kita membuka sebuah subtunnel di tunnel BGP ASnumber 100 yang sebelumnya sudah ada, yaitu subtunnel untuk "melewatkan" routing table vrf vpn1 dan vrf vpn2. Perhatikan bahwa untuk mengaktifkan "sub-tunnel" tersebut kita pakai command *neighbor <IP\_Address> activate*. Perhatikan juga bahwa MB-BGP

memanfaatkan fasilitas *community* pada standar BGP, baik yang *standard* maupun *extended*, yang kedua nya – standard atau extended (*both*) saling dipertukarkan.

Setelah subtunnel itu terbentuk, Anda bisa memeriksa statusnya, dengan:

```
PE1# sh ip bgp neigh
```

Anda akan melihat pada bagian address-family: vpn4 sudah aktif atau belum. Dan Anda juga bisa melihat apakah ada prefix yang sudah diterima oleh PE1 atau dikirimkan oleh PE1. Jika Anda berhasil menemukan hal ini, maka berarti langkah Anda sudah benar. Dan perlu dicatat bahwa saat Anda memasukkan dan mengaktifkan MP-BGP diatas, maka konfigurasi untuk persiapan “menyuntikkan” routing prefix untuk vrf vpn1 dan vrf vpn2 langsung terbentuk, sbb:

```
PE1# sh run
address-family ipv4 vrf vpn2
no synchronization
exit-address-family
!
address-family ipv4 vrf vpn1
no synchronization
exit-address-family
!
```

Konfigurasi diatas tidak Anda ketik manual, tetapi otomatis ditambahkan oleh Cisco IOS.

Nah jika Anda sudah menemukan konfigurasi tsb, ini adalah langkah terakhir yang perlu Anda lakukan. Yaitu, kita akan me-nyuntik-kan routing table vrf vpn1 yang static dan connected.

Caranya seperti ini.

```
PE1(config)#router bgp 100
PE1(config-router)#address-family ipv4 vrf vpn1
PE1(config-router-a)# redistribute connected
PE1(config-router-a)# redistribute static
```

Sehingga secara lengkap konfigurasi BGP + MP-BGP untuk router PE1, adalah:

```
PE1# sh run
router bgp 100
no synchronization
bgp log-neighbor-changes
neighbor 192.168.100.3 remote-as 100
neighbor 192.168.100.3 update-source Loopback0
no auto-summary
!
address-family vpnv4
```

```

neighbor 192.168.100.3 activate
neighbor 192.168.100.3 send-community both
exit-address-family
!
address-family ipv4 vrf vpn2
redistribute connected
redistribute static
no synchronization
exit-address-family
!
address-family ipv4 vrf vpn1
redistribute connected
redistribute static
no synchronization
exit-address-family
!

```

Demikian juga hal dan tahapan yang sama Anda lakukan untuk PE2, sehingga secara lengkap konfigurasi BGP + MP-BGP untuk router PE2, adalah:

```

PE2# sh run
router bgp 100
no synchronization
bgp log-neighbor-changes
neighbor 192.168.100.1 remote-as 100
neighbor 192.168.100.1 update-source Loopback0
no auto-summary
!
address-family vpnv4
neighbor 192.168.100.1 activate
neighbor 192.168.100.1 send-community both
exit-address-family
!
address-family ipv4 vrf vpn2
redistribute connected
redistribute static
no synchronization
exit-address-family
!
address-family ipv4 vrf vpn1
redistribute connected
redistribute static
no synchronization
exit-address-family
!

```

Setelah Anda mengaktifkan MP-BGP, dan menyuntikkan routing table PE1 vrf vpn1 dan routing table PE1 vrf vpn2 kedalam MP-BGP, Anda bisa memeriksa apakah prefix yang Anda suntikkan tersebut diterima di PE2. Anda bisa memeriksa dengan:

```
PE2# sh ip bgp neigh
```

Pada bagian address-family: vpnv4, Anda akan bisa melihat bahwa ada beberapa prefix yang di receive oleh PE2 yang di kirim oleh PE1. Disamping itu ada juga informasi bahwa ada beberapa prefix yang dikirim oleh PE2 ke PE1.

Jika Anda berhasil dengan langkah KETUJUH ini maka langkah terakhir adalah, mengaktifkan dan meng-configure router VPN1\_A, dan VPN1\_X yang terhubung ke PE1. Setelah itu Anda aktifkan dan configure router VPN1\_B, dan VPN1\_Y, yang terhubung ke PE2.

### **KEDELAPAN. MENGAKTIFKAN CE dan MENGUJI LINK END-To-END**

Setelah selesai dengan langkah KETUJUH, maka pada langkah ini, Anda harus mengaktifkan dan meng-configure router VPN1\_A, dan VPN1\_X yang terhubung ke PE1. Setelah itu Anda aktifkan dan configure router VPN1\_B, dan VPN1\_Y, yang terhubung ke PE2.

Konfigurasinya ke-empat router CE tersebut sangat sangat sederhana, karena Anda tidak perlu meng-configure MPLS, BGP, MB-BGP, dan VRF. Silahkan lihat pada bagian Appendix untuk konfigurasi lengkap 4 router CE tsb.

Setelah itu, pekerjaan paling final, adalah memeriksa apakah router VPN1\_A bisa PING ke lawannya yaitu router VPN1\_B, dan sebaliknya? Contohnya:

```
VPN1_A# ping 202.147.192.2  
VPN1_B# ping 202.147.192.1
```

Jika sukses maka berarti Anda sudah berhasil mensimulasikan jaringan MPLS-L3VPN end to end. Dan jika Anda ingin yakin bahwa paket PING tsb dibalas oleh “lawan” yang benar, Anda bisa mencoba telnet:

```
VPN1_A# telnet 202.147.192.2
```

Jika kemudian Anda terhubung dengan router VPN1\_B (bukan VPN2\_Y), maka... Selamat!!! Anda telah berhasil mensimulasikan sebuah jaringan MPLS-L3VPN.

## Biografi Penulis



**Rafdian Rasyid.** Menyelesaikan S1 bidang *Computer Networking* pada jurusan Teknik Elektro ITB tahun 1995, dan S2 bidang *Computer Science* pada Fakultas Ilmu Komputer UI tahun 2001 (*cum-laude*). Penulis memiliki kegemaran di bidang jaringan komputer, dan sebelumnya memiliki sertifikasi di bidang ini seperti: *CCNA, CCNP, Certified Unix Administrator, Certified Internet Security Specialist*. Disamping itu penulis telah mengikuti berbagai macam pelatihan di bidang komunikasi satelit, pelatihan *management* dan *leadership*.

Penulis saat ini bekerja sebagai praktisi dibidang ICT.

## APPENDIX – KONFIGURASI LENGKAP SEMUA ROUTER

Catatan: abaikan perintah `half duplex` pada semua interface Ethernet. Hal ini disebabkan by default IOS Cisco 3640 yang saya gunakan, membuat konfigurasi untuk interface Ethernet, secara default: duplex half. Oleh karena fokus kita bukan pada Ethernet speed dan mode duplex, maka saya biarkan Ethernet adapter bekerja dalam modus yang otomatis diset IOS yaitu half duplex.

### KONFIGURASI

```
PE1
----- start PE1
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname PE1
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
memory-size iomem 5
!
!
ip cef
!
!
ip vrf vpn1
  rd 100:1
  route-target export 100:1
  route-target import 100:1
!
ip vrf vpn2
  rd 100:2
  route-target export 100:2
  route-target import 100:2
!
mpls label protocol ldp
!
!
!
!
interface Loopback0
```

```

    ip address 192.168.100.1 255.255.255.255
    !
interface Ethernet0/0
    description *** Connection to Core ***
    ip address 192.168.2.1 255.255.255.0
    half-duplex
    mpls ip
    !
interface Ethernet0/1
    description *** Connection to CE1 ***
    ip vrf forwarding vpn1
    ip address 192.168.1.2 255.255.255.0
    half-duplex
    !
interface Ethernet0/2
    description *** Connection to VPN2_X ***
    ip vrf forwarding vpn2
    ip address 192.168.6.2 255.255.255.0
    half-duplex
    !
interface Ethernet0/3
    no ip address
    shutdown
    half-duplex
    !
router ospf 100
    log-adjacency-changes
    network 192.168.2.0 0.0.0.255 area 100
    network 192.168.100.1 0.0.0.0 area 100
    !
router bgp 100
    no synchronization
    bgp log-neighbor-changes
    neighbor 192.168.100.3 remote-as 100
    neighbor 192.168.100.3 update-source Loopback0
    no auto-summary
    !
    address-family vpnv4
    neighbor 192.168.100.3 activate
    neighbor 192.168.100.3 send-community both
    exit-address-family
    !
    address-family ipv4 vrf vpn2
    redistribute connected
    redistribute static
    no synchronization
    exit-address-family

```

```

!
address-family ipv4 vrf vpn1
redistribute connected
redistribute static
no synchronization
exit-address-family
!
no ip http server
ip route vrf vpn1 202.147.192.1 255.255.255.255 192.168.1.1
ip route vrf vpn2 202.147.192.1 255.255.255.255 192.168.6.1
!
!
!
!
mpls ldp router-id Loopback0 force
!
control-plane
!
!
line con 0
line aux 0
line vty 0 4
  login
!
end
----- end PE1

PE2
----- start PE2
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname PE2
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
memory-size iomem 5
!
!
ip cef
!

```

```

!
ip vrf vpn1
  rd 100:1
  route-target export 100:1
  route-target import 100:1
!
ip vrf vpn2
  rd 100:2
  route-target export 100:2
  route-target import 100:2
!
mpls label protocol ldp
!
!
!
!
interface Loopback0
  ip address 192.168.100.3 255.255.255.255
!
interface Ethernet0/0
  description *** Connection to Core ***
  ip address 192.168.3.2 255.255.255.0
  half-duplex
  mpls ip
!
interface Ethernet0/1
  description *** Connection to VPN1_B ***
  ip vrf forwarding vpn1
  ip address 192.168.4.1 255.255.255.0
  half-duplex
!
interface Ethernet0/2
  description *** Connection to VPN2_Y ***
  ip vrf forwarding vpn2
  ip address 192.168.5.1 255.255.255.0
  half-duplex
!
interface Ethernet0/3
  no ip address
  shutdown
  half-duplex
!
router ospf 100
  log-adjacency-changes
  network 192.168.3.0 0.0.0.255 area 100
  network 192.168.100.3 0.0.0.0 area 100
!

```

```

router bgp 100
  no synchronization
  bgp log-neighbor-changes
  neighbor 192.168.100.1 remote-as 100
  neighbor 192.168.100.1 update-source Loopback0
  no auto-summary
  !
  address-family vpnv4
  neighbor 192.168.100.1 activate
  neighbor 192.168.100.1 send-community both
  exit-address-family
  !
  address-family ipv4 vrf vpn2
  redistribute connected
  redistribute static
  no synchronization
  exit-address-family
  !
  address-family ipv4 vrf vpn1
  redistribute connected
  redistribute static
  no synchronization
  exit-address-family
  !
no ip http server
ip route vrf vpn1 202.147.192.2 255.255.255.255 192.168.4.2
ip route vrf vpn2 202.147.192.2 255.255.255.255 192.168.5.2
!
!
!
!
mpls ldp router-id Loopback0 force
!
control-plane
!
!
line con 0
line aux 0
line vty 0 4
  login
!
end
----- end PE2

Core
----- start Core
!

```

```

version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Core
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
memory-size iomem 5
!
!
ip cef
!
!
mpls label protocol ldp
!
!
!
!
interface Loopback0
 ip address 192.168.100.2 255.255.255.255
!
interface Ethernet0/0
 description *** Connection to PE1 ***
 ip address 192.168.2.2 255.255.255.0
 half-duplex
 mpls ip
!
interface Ethernet0/1
 description *** Connection to PE2 ***
 ip address 192.168.3.1 255.255.255.0
 half-duplex
 mpls ip
!
interface Ethernet0/2
 no ip address
 shutdown
 half-duplex
!
interface Ethernet0/3
 no ip address
 shutdown
 half-duplex

```

```

!
router ospf 100
  log-adjacency-changes
  network 192.168.2.0 0.0.0.255 area 100
  network 192.168.3.0 0.0.0.255 area 100
  network 192.168.100.2 0.0.0.0 area 100
!
no ip http server
!
!
!
!
mpls ldp router-id Loopback0 force
!
control-plane
!
!
line con 0
line aux 0
line vty 0 4
  password cisco
  login
!
end
----- end Core

```

```

VPN1_A
----- start VPN1_A

```

```

version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname VPN1A
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
memory-size iomem 5
!
!
ip cef
!
!

```

```

!
!
!
!
interface Loopback0
 ip address 202.147.192.1 255.255.255.255
!
interface Ethernet0/0
 description *** Connection to PE1 ***
 ip address 192.168.1.1 255.255.255.0
 half-duplex
!
interface Ethernet0/1
 no ip address
 shutdown
 half-duplex
!
interface Ethernet0/2
 no ip address
 shutdown
 half-duplex
!
interface Ethernet0/3
 no ip address
 shutdown
 half-duplex
!
no ip http server
ip route 202.147.192.2 255.255.255.255 192.168.1.2
ip route 192.168.4.0 255.255.255.0 192.168.1.2
!
!
!
!
!
control-plane
!
!
line con 0
line aux 0
line vty 0 4
!
end
----- end VPN1_A

VPN2_X
----- start VPN2_X

```

```

!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname VPN2_X
!
!
memory-size iomem 15
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
ip subnet-zero
!
!
!
ip audit notify log
ip audit po max-events 100
ip ssh time-out 120
ip ssh authentication-retries 3
!
!
!
!
interface Loopback0
 ip address 202.147.192.1 255.255.255.255
!
interface Ethernet0/0
 description *** Connection to PE1 ***
 ip address 192.168.6.1 255.255.255.0
 half-duplex
!
interface FastEthernet0/0
 no ip address
 shutdown
 speed auto
!
ip classless
ip route 202.147.192.2 255.255.255.255 192.168.6.2
ip route 192.168.5.0 255.255.255.0 192.168.6.2
no ip http server
ip pim bidir-enable
!
!
!

```

```

!
line con 0
line aux 0
line vty 0 4
!
no scheduler allocate
end
----- end VPN2_X

VPN1_B
----- start VPN1_B
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname VPN1B
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
memory-size iomem 5
!
!
ip cef
!
!
!
!
!
interface Loopback0
 ip address 202.147.192.2 255.255.255.255
!
interface Ethernet0/0
 description ** Connection to PE2 ***
 ip address 192.168.4.2 255.255.255.0
 half-duplex
!
interface Ethernet0/1
 no ip address
 shutdown
 half-duplex
!

```

```

interface Ethernet0/2
  no ip address
  shutdown
  half-duplex
!
interface Ethernet0/3
  no ip address
  shutdown
  half-duplex
!
ip http server
ip route 0.0.0.0 0.0.0.0 192.168.4.1
!
!
!
!
!
control-plane
!
!
line con 0
line aux 0
line vty 0 4
!
end
----- end VPN1_B

VPN2_Y
----- start VPN2_Y
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname VPN2_Y
!
!
memory-size iomem 15
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
ip subnet-zero
!
!
!

```

```

ip audit notify log
ip audit po max-events 100
ip ssh time-out 120
ip ssh authentication-retries 3
!
!
!
!
interface Loopback0
 ip address 202.147.192.2 255.255.255.255
!
interface Ethernet0/0
 description *** Connection to PE2 ***
 ip address 192.168.5.2 255.255.255.0
 half-duplex
!
interface FastEthernet0/0
 no ip address
 shutdown
 speed auto
!
ip classless
ip route 0.0.0.0 0.0.0.0 192.168.5.1
no ip http server
ip pim bidir-enable
!
!
!
!
line con 0
line aux 0
line vty 0 4
!
no scheduler allocate
end
----- end VPN2_Y

```